

What is claimed:

1. A computer architecture for sharing information between plural applications having disparate data structures, said architecture comprising:

plural applications, at least one of said applications having a data structure that is different from another of said applications;

an application integration platform including logic for exchanging information between said plural applications; and

at least one common object definition specifying common objects to be used for exchanging data between said applications and including a canonical object defining elements of a standard object that are common between data structures of said plural applications, said common object further including at least one extension defining application specific or user specific elements, said canonical object being exposed to all of the applications through said application integration platform, said extension being exposed only to selected ones of the plural applications.

2. A computer architecture as recited in claim 1, wherein said at least one extension comprises an application specific extension having data elements used only by a first of said plural applications and a user specific extension having data elements not in said canonical object but desired by a specific user.

3. A computer architecture as recited in claim 1, wherein said common object definition comprises a tree like structure.

4. A computer architecture as recited in claim 3, wherein each of said canonical object and said extensions are represented by a separate node in said common object definition.

5. A computer architecture as recited in claim 3, wherein each of said canonical object and said extensions are represented by a distinct DTD in said common object definition.
6. A computer architecture as recited in claim 3, wherein said common object definition references another common object definition.
7. A computer architecture as recited in claim 1, further comprising means for cross referencing data elements in said common object definition with corresponding data elements in said applications.
8. A computer architecture as recited in claim 1, wherein said application integration platform is operative to enforce plural system of record policies.
9. A computer architecture as recited in claim 8, wherein said system of record policies include a federated policy in which different ones of said applications is responsible for updating different portions of common business objects corresponding to a particular common business object definition.
10. A computer architecture as recited in claim 8, wherein said system of record policies include a revolving policy in which different ones of said applications is responsible for updating common business objects corresponding to a particular common business object definition at different points of the life cycle of the common business object.
11. A computer architecture as recited in claim 8, wherein said system of record policies include a rules based policy in which common business objects corresponding to a particular common business object definition are updated in different manners based on external factors applied to predetermined rules.
12. A computer architecture as recited in claim 8, wherein said system of record policies include a rules based policy in which common business objects are updated based on external factors as applied to predetermined rules.

13. A computer architecture as recited in claim 1, wherein said integration platform comprises at least one connector having a transformation map, said transformation map comprising plural map modules applied in seriatim.

14. A computer architecture as recited in claim 13, wherein said plural map modules comprise a first map module having a data map for the canonical object, a second map module having a data map for a user extension and a third map module having a data map for an application extension.

15. A method of defining a common data object for sharing information between plural applications having disparate data structures, said method comprising:

identifying one or more primary applications each having a data structure;

determining common data elements between the data structures;

selecting elements of a canonical object that correspond to the common elements;

adjusting the canonical object based on a common object standard;
and

adding at least one application specific or user specific extension to the data elements of the canonical object.

16. The method as recited in claim 15, wherein said adding step comprises adding data elements of a specified application to maintain functionality of the specified application in a system using the common object.

17. The method as recited in claim 15, wherein said adding step comprises adding data elements of one or more applications to maintain functionality desired by specified users in a system using the common object.

18. A common object definition for common objects used for sharing information between plural applications having disparate data structures, said definition comprising:

a canonical object defining elements of a standard object that are common between data structures of said plural applications; and

at least one extension defining application specific or user specific elements, said canonical object being exposed to all of the applications, said extension being exposed only to selected ones of the plural applications.

19. A definition as recited in claim 18, wherein said at least one extension comprises an application specific extension having data elements used only by a first of plural applications and a user specific extension having data elements not in said canonical object but desired by a specific user.

20. A definition as recited in claim 18, wherein said canonical object and said extension are defined by a tree like structure.

21. A computer architecture as recited in claim 20, wherein each of said canonical object and said extensions are represented by a separate node.

22. A computer architecture as recited in claim 20, wherein each of said canonical object and said extensions are represented by a distinct DTD.

23. A computer architecture as recited in claim 20, wherein said common object definition references another common object definition.